REMARKS

Applicant has studied the Office Action dated October 19, 2006. It is submitted that the application is in condition for allowance. Claims 1-7, 9-14, 16, 17, 19-22, and 24-28 are pending. Reconsideration and allowance of the pending claims in view of the following remarks are respectfully requested.

Claims 1, 14, and 22 were rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1 and 16 of U.S. Patent No. 7,116,783. This rejection is respectfully traversed.

In claims 1 and 16 of the '783 patent, during the data transfer of an N-byte data element, each byte of the data element is transferred at least one time on the data bus, and may be transferred two or more times. The number of times one byte of the data element is transferred on the data bus is random, and at the end of the data transfer the total number of bytes that have been transferred is also random. In contrast, in claims 1, 14, and 22 of the present application, during the data transfer of an N-byte data element, each byte of the data is transferred exactly one time on the data bus. At the end of the data transfer, N total bytes of data have always been transferred.

Additionally, in claims 1 and 16 of the '783 patent, before each transfer of a byte of the N-byte data element, the place value of the byte to be transferred is randomly chosen. In contrast, in claims 1, 14, and 22 of the present application, a transfer rule defines the order in which the bytes of the N-byte data element are successively transferred through the data bus. At least one parameter of the transfer rule is randomly chosen before the transfer of the (complete) N-byte data element, and this same value is used for that parameter of the transfer rule for all bytes of the N-byte data element. Further, in claims 1 and 16 of the '783 patent, the transfer of the N-byte data element ends when a loading indicator takes a predetermined value. Thus, the duration of the transfer is random. In contrast, in claims 1, 14, and 22 of the present application, the transfer of the N-byte data element ends when N bytes have been transferred. Thus, the duration of the transfer is always the same.

Therefore, claims 1, 14, and 22 of the present application distinguish over claims 1 and 16 of the '783 patent, and the double patenting rejection of these claims should be withdrawn.

Claims 1, 2, 14, 22, and 25-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Iida (U.S. Patent No. 5,422,727) in view of Pfab (U.S. Patent No. 6,195,752). Claims 3-7, 9-13, 16, 17, 19-21, and 24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Iida in view of Pfab and Menezes et al. ("Book of Applied Cryptography"). These rejections are respectfully traversed.

The present invention is directed to methods and circuits for transferring data in a highly secure manner. One preferred embodiment provides a method for secured transfer through a data bus that is connected between a first memory and a second memory. According to the method, an N-byte data element is provided in the first memory, and the value of at least one parameter of a transfer rule is randomly chosen before a transfer of the N-byte data element. The transfer rule defines the order in which the bytes of the N-byte data element are successively transferred through the data bus. The N bytes of the data element are successively transferred byte-by-byte through the data bus to the second memory in the order specified by the transfer rule, with each of the N bytes transiting once and only once through the data bus.

Thus, in this transfer method, the bytes of the data element stored in a first memory are successively transferred byte-by-byte to the second memory, and the transfer rule with at least one randomly chosen parameter is used to select the byte to be transferred at each successive transfer of a byte. The transfer rule, randomly chosen parameter(s), and successive byte-by-byte transfer of the data element operate together so that the N bytes of the data element are not transferred in the same successive byte order for each transfer of that data element. Accordingly, the simple power analysis method of snooping is not sufficient to obtain the value of the N-byte data element transiting through the data bus.

The Iida reference is directed to a facsimile machine that uses a magnetic tape and a buffer memory for data storage and transfer. The Pfab reference is directed to a data processing

circuit in which the data transiting on a data bus and stored in memory is encoded. However, neither Pfab nor Iida, or a combination of the two, discloses a method for secured transfer in which the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are successively transferred, is randomly chosen before a transfer of the data element, and the N bytes of the data element are successively transferred byte-by-byte through a data bus in the order specified by the transfer rule, as is recited in independent claim 1. Independent claim 14 contains similar recitations.

Likewise, neither Pfab nor Iida, or a combination of the two, discloses a programmable circuit that includes a random number generator that before a transfer supplies the value of at least one parameter of a data transfer rule that defines the order in which the bytes of an N-byte data element are successively transferred, and a control unit that controls a data bus such that the N bytes of the data element are successively transferred byte-by-byte through the data bus in the order specified by the data transfer rule, as is recited in independent claim 22.

As recognized by the Examiner, Iida does not disclose randomly choosing the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are successively transferred through the data bus, before a transfer of the N-byte data element. However, the Examiner has taken the position that the Pfab reference makes up for this deficiency in the disclosure of Iida by disclosing such a feature. This position of the Examiner is respectfully traversed.

In embodiments of the present invention, the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte through a data bus, is randomly chosen before a successive byte-by-byte transfer of the data element through a data bus in the byte order that is specified by the transfer rule. More specifically, a transfer rule defines the order in which the N bytes of an N-byte data element are successively transferred through a data bus, and the value of one or more parameters of the transfer rule are randomly chosen before a transfer of the data element. The N bytes of the data element are successively transferred byte-by-byte through the data bus in the byte order

specified by the transfer rule, with each of the N bytes transiting once and only once through the data bus.

Thus, the bytes of the N-byte data element are successively transferred byte-by-byte through the data bus, and the transfer rule with at least one randomly chosen parameter is used to select the byte to be transferred at each successive transfer of one byte of the N bytes of the data element. Thus, the N bytes of the data element are not successively transferred in the same order for each transfer of that data element.

In contrast, Pfab discloses a data processing circuit in which data is stored in memory and transferred through the data bus in an encoded format. In the first and second embodiments, the data processing circuit includes a microprocessor 101, a data bus 106, and memories 102-105, as shown in Figures 1 and 2. Each memory 102-105 stores encoded data, and this encoded data is transferred through the data bus 106. The microprocessor 101 includes an encoding module 107 that decodes the encoded data received from the data bus 106, and encodes data to be sent each memory 102-105 through the data bus 106.

In the third embodiment, the data processing circuit includes a microprocessor 1, data buses 6-15, and memories 2-5, as shown in Figure 3. Each memory 2-5 stores encoded data, and this data is transferred through the data buses 6-15 at least partially encoded. The microprocessor 1 includes one encoding module 35 and an additional encoding module 20-22 is provided on the data buses 6-15 between each memory 2-5 and the microprocessor 1. The encoded data stored in each memory 2-5 is partially decoded by the associated encoding module 20-22 and then completely decoded by the encoding module 35 of the microprocessor. Similarly, data sent to each memory 2-5 is partially encoded by the encoding module 35 of the microprocessor and then completely encoded by the associated encoding module 20-22. Thus, Pfab teaches data processing circuits in which data is protected by modifying (i.e., encoding) each byte of data stored in memory and transferred through the data bus.

While Pfab teaches modifying each data unit (e.g., byte or "word") that is transferred through the data bus at one time, Pfab does not teach or suggest modifying the order in which

these data units (e.g., bytes or "words") are successively transferred through the data bus. In other words, Pfab teaches altering each data unit that transits through the data bus, but does not teach or suggest altering the order in which multiple data units are successively transferred data unit-by-data unit through the data bus.

The Examiner states that "Pfab's disclosure of altering the significance of the different bits is equivalent to that of defining the order in which the bytes are transferred because of the fact that 1 byte consists of eight bits." Pfab discloses that an encoding module encodes data traffic on the data bus, and states that the encoding can be performed by altering the significance of individual data bits. See Pfab at 6:44-57. Pfab teaches using hardware encoding to alter the significance of individuals bits by changing "low" bits of a data unit to be transferred to "high" bits when the data unit is transferred on the data bus. See Pfab at 3:40-49.

Thus, Pfab's disclosure of "altering the significance of individual data bits" entails changing the bit values of each data unit that transits through the data bus. In other words, the values of bits in one data unit (i.e., the amount of data that is transported simultaneously on the data bus) are modified. Pfab's disclosure of "altering the significance of individual data bits" does not teach or suggest changing the order in which mulitple data units are successively transferred data unit-by-data unit through the data bus.

In contrast, in embodiments of the present invention, the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte through a data bus, is randomly chosen before a successive byte-by-byte transfer of the data element through a data bus in the byte order that is specified by the transfer rule. More specifically, the claims recite a data transfer having the features of:

1) successive byte-by-byte transfer of an N-byte data element (i.e., one byte of the data element is transferred, then another byte, and so on);

2) the order in which the bytes of the element data are successively transferred is defined by a transfer rule having randomly chosen parameter(s); and

3) each byte of the N-byte data element is transfered once and only once during the transfer of the N-byte data element.

This successive transfer of the N bytes of the data element byte-by-byte through the data bus in a byte order that is specified by a transfer rule having randomly chosen parameter(s) is used to alter the order in which data units (i.e., bytes in this case) are transferred data unit-by-data unit through the data bus. This is fundamenmtally different than Pfab's teaching of altering the values of the bits in each data unit that is transferred through the data bus.

Further, "the fact that 1 byte consists of eight bits" is irrelevant to the fundamental difference between the transfer method taught in Pfab and the transfer method recited in the claims. Pfab teaches altering each data unit before that data unit is transferred on the data bus, while the claims recite altering the order in which multiple data units are successively transferred data unit-by-data unit through the data bus. Regardless of the number of bits or bytes in a data unit, Pfab only teaches altering each data unit that is successively transferred on the data bus. Pfab does not teach or suggest changing the order in which data units are successively transferred data unit-by-data unit through a data bus. In other words, if an N-byte data element is transferred byte-by-byte (as recited in the claims) using the transfer method of Pfab, each byte that is transferred will be changed, but the N bytes will always be transferred in the same byte order.

The Examiner also states that "Pfab's disclosure of determining which bit lines should be used also determines the byte order in which the N-byte data element will progress on the bus." Pfab never teaches "determining which bit lines should be used" as stated by the Examiner. In Pfab, all of the bit lines of the data bus are always used.

Pfab actually teaches that an encoding module can encode data traffic on a data bus by interchanging individual bit lines of the data bus. See Pfab at 6:44-57. Pfab's disclosure of "interchanging individual data bits" entails changing the order of the bits of each data unit when that data unit transits through the data bus. In other words, the bit order within one data unit (i.e., the amount of data that is transported simultaneously on the data bus) is modified. Pfab's disclosure of "interchanging individual data bits" does not teach or suggest changing the order in which mulitple data units are successively transferred data unit-by-data unit through the data bus.

In contrast, in embodiments of the present invention, the value of at least one parameter of a transfer rule, which defines the order in which the bytes of an N-byte data element are

successively transferred byte-by-byte through a data bus, is randomly chosen before a successive byte-by-byte transfer of the data element through a data bus in the byte order that is specified by the transfer rule. This successive transfer of the N bytes of the data element byte-by-byte through the data bus in a byte order that is specified by a transfer rule having randomly chosen parameter(s) is used to alter the order in which data units (i.e., bytes in this case) are transferred data unit-by-data unit through the data bus. This is fundamenmtally different than Pfab's teaching of altering the bit order in each data unit that is transferred through the data bus.

Pfab teaches altering each data unit that is transferred on the data bus, while the claims recite altering the order in which multiple data units are successively transferred data unit-by-data unit through the data bus. Pfab does not teach or suggest changing the order in which data units are successively transferred data unit-by-data unit through a data bus. In other words, if an N-byte data element is transferred byte-by-byte (as recited in the claims) using the transfer method of Pfab, each byte that is transferred will be changed, but the N bytes will always be transferred in the same byte order.

The Examiner also states that randomly selecting a key in the transfer method Pfab is equivalent to randomly choosing a parameter of the transfer rule. Pfab teaches using an encoding module to encode data traffic on a data bus by changing each data unit that transits through the data bus. Pfab also teaches randomly selecting a key with which to encode each data unit. In other words, the key for encoding each individual data unit (i.e., the amount of data that is transported simultaneously on the data bus) is randomly selected.

In contrast, in embodiments of the present invention, there is randomly selected the value of at least one parameter of a transfer rule that defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte through a data bus, before a successive byte-by-byte transfer of the data element through a data bus in the byte order that is specified by the transfer rule. The randomly chosen parameter(s) of the transfer rule is used to alter the order in which data units (i.e., bytes in this case) are transferred data unit-by-data unit through the data bus. This is fundamenmtally different than Pfab's teaching of randomly choosing a key for altering each data unit that is transferred through the data bus.

Pfab teaches randomly choosing a key for altering each data unit that is transferred on the data bus, while the claims recite randomly choosing a parameter(s) of the transfer rule for altering the order in which multiple data units are successively transferred data unit-by-data unit through the data bus. Pfab does not teach or suggest randomly choosing a parameter(s) of a transfer rule for changing the order in which data units are successively transferred data unit-by-data unit through a data bus. In other words, if an N-byte data element is transferred byte-by-byte (as recited in the claims) using the transfer method of Pfab, each byte that is transferred will be changed, but the N bytes will always be transferred in the same byte order regardless of the randomly selected key.

The Examiner also states that "Pfab teaches that the data element is transferred byte-by-byte because a sequence of 8 bits is equivalent to one byte and there is an operating module that can influence the encoding using different conversion methods." As explained above, Pfab teaches using an encoding module to encode data traffic on a data bus by changing each data unit that transits through the data bus. Pfab teaches different methods for altering each data unit, such as using an encoding key to encode each data unit, changing bit values within each data unit, and changing the order of the bits of the data unit. However, all of the encoding or conversion methods disclosed in Pfab entail changing each individual data unit (i.e., the amount of data that is transported simultaneously on the data bus).

In contrast, in embodiments of the present invention, a transfer rule defines the order in which the bytes of an N-byte data element are successively transferred byte-by-byte through a data bus, and the data element is successively transferred byte-by-byte through a data bus in the byte order that is specified by the transfer rule. The transfer rule is used to alter the order in which data units (i.e., bytes in this case) are transferred data unit-by-data unit through the data bus. This is fundamenmtally different than Pfab's teaching of altering each data unit that is transferred through the data bus.

Further, the fact that eight bits is equal to one byte is irrelevant to the fundamental difference between the transfer method taught in Pfab and the transfer method recited in the claims. Pfab teaches altering each data unit before that data unit is transferred on the data bus,

while the claims recite altering the order in which multiple data units are successively transferred data unit-by-data unit through the data bus. Regardless of the number of bits or bytes in a data unit, Pfab only teaches altering each data unit that is successively transferred on the data bus. Pfab does not teach or suggest changing the order in which data units are successively transferred data unit-by-data unit through a data bus. In other words, if an N-byte data element is transferred byte-by-byte (as recited in the claims) using the transfer method of Pfab, each byte that is transferred will be changed, but the N bytes will always be transferred in the same byte order.

Pfab teaches altering each data unit before that data unit is transferred on the data bus, while the claims recite altering the order in which multiple data units are successively transferred data unit-by-data unit through the data bus. In Pfab, a data element having four data units D1-D4 will always be successively transferred data unit-by- data unit in the same order: D1, D2, D3, then D4. Pfab secures the data element by altering each of the four data units Dx, not by changing the order in which these data units are transferred. In embodiments of the present invention, the four data units D1-D4 of the data element are not always successively transferred through the data bus in the same order. Instead, the data units are successively transferred data unit-by- data unit through the data bus in an order defined by a transfer rule (with one or more randomly chosen parameters). Thus, the data element may be successively transferred data unit-by- data unit in the order D3, D2, D1, then D4, then that same data element may be successively transferred data unit-by- data unit in the order D2, D3, D4, then D1, and then that same data element may be successively transferred data unit-by- data unit in the order D4, D3, D2, then D1.

Pfab teaches randomly choosing a key or using some other method for altering each data unit that is transferred on a data bus. Pfab does not teach or suggest randomly choosing a parameter(s) of a transfer rule for changing the order in which data units are successively transferred data unit-by-data unit through a data bus. Applicant believes that the differences between Iida, Pfab, and the present invention are clear in claims 1, 14, and 22, which set forth various embodiments of the present invention. Therefore, claims 1, 14, and 22 distinguish over the Iida and Pfab references, and the rejection of these claims under 35 U.S.C. § 103(a) should be withdrawn.

As discussed above, claims 1, 14, and 22 distinguish over the Iida and Pfab references. Furthermore, the claimed features of the present invention are not realized even if the teachings of Menezes are incorporated into Iida and Pfab. Menezes does not teach or suggest the claimed features of the present invention that are absent from Iida and Pfab. Thus, claims 1, 14, and 22 distinguish over the Iida, Pfab, and Menezes references, and thus, claims 2-7, 9-13, and 25-28, claims 16, 17, and 19-21, and claim 24 (which depend from claims 1, 14, and 22, respectively) also distinguish over the Pfab reference. Therefore, it is respectfully submitted that the rejections of claims 1-7, 9-14, 16, 17, 19-22, and 24-8 under 35 U.S.C. § 103(a) should be withdrawn.

Applicant has examined the reference cited by the Examiner as pertinent but not relied upon. It is believed that this reference neither discloses nor makes obvious the invention recited in the present claims. In view of the foregoing, it is respectfully submitted that the application and the claims are in condition for allowance. Reexamination and reconsideration of the application are requested.

If for any reason the Examiner finds the application other than in condition for allowance, the Examiner is invited to call the undersigned attorney at (561) 989-9811 should the Examiner believe a telephone interview would advance the prosecution of the application.

Respectfully submitted,

Date: April 19, 2007

By: /Stephen Bongini/
Stephen Bongini
Registration No. 40,917
Attorney for Applicant

FLEIT KAIN GIBBONS
GUTMAN BONGINI & BIANCO P.L.
One Boca Commerce Center
551 Northwest 77th Street, Suite 111
Boca Raton, Florida 33487
Telephone: (561) 989-9811
Facsimile: (561) 989-9812